

UNITED STATES PATENT APPLICATION

FOR

**METHOD AND APPARATUS FOR INCREASING PROCESSING PERFORMANCE OF
PIPELINED AVERAGING FILTERS**

INVENTOR:

Jonathan Lamb, a citizen of the United Kingdom

ASSIGNED TO:

Semtech Corporation, a Delaware Corporation

PREPARED BY:

**THELEN, REID, & PRIEST, LLP
P.O. BOX 640640
SAN JOSE, CA 95164-0640
TELEPHONE: (408) 292-5800
FAX: (408) 287-8040**

Attorney Docket Number: SEMT-024

S P E C I F I C A T I O N

TITLE OF THE INVENTION

Method And Apparatus For Increasing Processing Performance
of Pipelined Averaging Filters

REFERENCE TO THE PROVISIONAL APPLICATION

[0001] This application claims the benefit of the United States Provisional Patent Application Serial No. 60/287,229, filed on April 27, 2001.

FIELD OF THE INVENTION

[0001] The present invention is generally directed to computer pipelines. More specifically, the present invention is directed to increasing the processing performance of a pipelined averaging filter.

BACKGROUND OF THE INVENTION

[0001] A technique of processing pipelines to increase data throughput has long been known in the art. A long task is divided into components and each component is distributed to one processor. A new task can begin even though the former tasks have not been completed. In the pipelined operation, different components of the different tasks are executed at the same time by different processors. Presently, pipelines are in widespread use in nearly all types of data processing electronic equipment, such as sophisticated supercomputers, in which fast and efficient processing of data is essential to the overall operation of the system.

[0001] Pipelines have been developed in a wide variety of electronic manufacturing and circuit design configurations. One example of the use of a pipeline is an averaging filter used in digital signal processing. An averaging filter generally consists of at least one subtractor module in series with at least one adder module. Each subtractor and adder module typically has numerous adder logic units and data registers. In order to increase the processing efficiency and speed in each of the subtractor and the adder modules, their respective internal adder logic units and registers are typically placed in pipelined arrangements. While an effective approach for increasing processing efficiency and speed, the typical pipelined configuration is not without shortcomings in other aspects. These shortcoming are even more apparent in high speed averaging filters which operate at high clock rates.

SUMMARY OF THE INVENTION

[0001] A pipelined processor such as an averaging filter including at least one subtractor section and at least one adder section is disclosed. Both of the subtractor section and the adder section have a plurality of adder logic units. In comparison to the conventional processor, the processor of the present invention is streamlined by the application of one or more of three techniques. First, there is the interleaving approach where the subtractor section and the adder section are interleaved with one another. Second, there is the one delay feedback approach where the adder section includes a one delay feedback for each of the adder logic units. Third, there is the delay enable signal output approach where the averaging filter includes a delay enable signal output for each of the adder logic units of the adder section.

BRIEF DESCRIPTION OF THE DRAWINGS

[0001] The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more exemplary embodiments of the present invention, and together with the detailed description, serve to explain the principles and exemplary implementations of the invention.

[0001] In the drawings:

FIG. 1 is a circuit block diagram of an adder module according to the prior art;

FIG. 2 is a circuit block diagram of an exemplary embodiment of a first order averaging filter in accordance with the present invention; and

FIG. 3 is a timing diagram illustrating six delay enable signals in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0001] Various exemplary embodiments of the present invention are described herein in the context of methods and apparatus for increasing the processing performance of pipelined averaging filters. Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. Reference will now be made in detail to exemplary implementations of the present invention as illustrated in the accompanying drawings. The same reference indicators will be used throughout the drawings and the following detailed descriptions to refer to the same or like parts.

[0001] In the interest of clarity, not all of the routine features of the exemplary implementations described herein are shown and described. It will of course, be appreciated that in the development of any such actual implementation, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, such as compliance with application- and business-related constraints, and that these specific goals will vary from one implementation to another and from one developer to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art having the benefit of this disclosure.

[0001] Turning first to FIG. 1, a circuit block diagram of an adder module 100 according to the prior art is shown. As noted above, an averaging filter includes at least one subtractor module

and at least one adder module in series. Generally, the adder module has more components than the subtractor module. The single adder module 100 is shown as an example of the magnitude of the component numbers in a typical averaging filter. For clarity, the subtractor module is not shown. Typical subtractor module and averaging filter designs are well known to one of ordinary skill in the art. For a first order averaging filter, there is one subtractor module and one adder module and they operate together to produce a numerical average of an inputted number sequence according to the following mathematical equation:

$$y(n) = ax(n) + (1-a)y(n-1) \quad \text{Eq. 1}$$

where $y(n)$ represents the filter output at time n , $ax(n)$ represents the adder component, $(1-a)$ represents the subtractor component, and $y(n-1)$ representing the delay element at time $n-1$.

[0001] For discussion purposes, the adder module 100 shown operates on thirty six bit numbers and may be used in a conventional thirty six bit averaging filter. The pipelining takes the form of splitting each add operation into smaller add blocks which can be completed within the clock period of the digital clock that drives the circuit. The faster the clock is the smaller the add blocks have to be. In this case, assume that the thirty six bit addition is broken into six blocks of six bit additions. Other combinations of blocks are also possible with multiples of two bits being preferred. The addition results are stored in registers or D-type flip flops (DFFs). As shown, the thirty six bit adder module 100 includes adding blocks in the form of six adder logic units a1-a6 and five associated carry delay gates c1-c5. The adder module 100 also includes three hundred delay elements in the form of fifty delayed flip flop gate icons D0-D50 where each of the fifty icons D0-D50 represent six actual DFFs.

[0001] In order to complete the addition process, more than one processing cycle is required. The thirty six bit number (bits 0-35) is first parsed into six segments of six bit lengths and entered into the adder module 100 via input gates 120-125. Each six bit segment is added in turn along parallel paths. In the first clock cycle, the adder logic unit a1 receives the first of six bit segments $x(5-0)$, performs the addition operation, and stores the result in DFF 10 with the carry going to carry delay gate c1. Simultaneously, the bits in each of the other five segments of the thirty six bit number are loaded into the DFFs 1, 3, 5, 7, 9, respectively. Subsequent additions of the next bit segments are completed in the subsequent clock cycles in a similar manner via adder logic units a2-a5. The addition result for each six bit segment is stored or accumulated until all of the six bit segments have been processed. For the first six bit segment, the stored result is passed down a chain of DFFs 10, 20, 29, 37, 44, 50 with each subsequent clock cycle. Finally, once all of the necessary operations are carried out, the results are outputted in the form of outputs 110-116.

[0001] The subtractor module that is necessary to the operation of an averaging filter is also made of adders and therefore also uses adding blocks as shown in FIG. 1. To make a subtractor module, all the bits of the number are first inverted and then added to an integer value of one. This is a commonly used process known as a twos complement. By contrast to the adder module, the corresponding subtractor module needs only half as many bits. Thus, in the example shown, only an eighteen bit subtractor module would be coupled to the thirty six bit adder module to form the averaging filter. Nevertheless, the subtractor module also contains a significant number of circuit elements depending on the application. In total, 387 DFFs would be needed to form the thirty six bit averaging filter using conventional designs.

[0001] Turning now to FIG. 2, a circuit block diagram of an exemplary embodiment of a first order averaging filter 200 in accordance with the present invention is shown. The averaging filter 200 includes a subtractor section 220 interleaved with an adder section 210. As above, for discussion purposes, the averaging filter 200 operates on a thirty six bit number. The subtractor section 220 includes four adder logic units s1-s4. The adder section 210 includes six adder logic units a1-a6.

[0001] As shown, the averaging filter 200 employs an interleaving approach. According to this approach, the adder logic units s1-s4 in the subtractor section 220 are interleaved by being coupled to a corresponding adder logic unit a1-a4 in the adder section 210. In this way an output of each of the adder logic units s1-s4 in the subtractor section 220 is inputted directly into an input of a corresponding adder logic unit in the adder section 210. One advantage of the interleaving approach is that each segment of the inputted data string which is processed by an adder logic unit in the subtractor section 220 is then processed by an adder logic unit in the adder section 210 without first having to await the processing completion of the entire string in the subtractor section 220. For example, segment x(5-0) is processed by the adder logic unit s1 in the subtractor section 220 and then by the adder logic unit a1 in the adder section 210 before segment x(18) is processed by the adder unit s4 in the subtractor section 220. By comparison to the sequential two module approach described with respect to FIG. 1, the foregoing interleaving approach reduces the number of DFFs needed for processing the output of the subtractor section 220 and the input of the adder section 210.

[0001] Also shown in FIG. 2, the adder section 210 employs a one delay feedback approach. According to this approach, the output of the adder logic unit is fed back to the input of the adder logic unit after being delayed for one clock cycle. For example, the $x(5-0)$ output of the adder logic unit a1 is fed back to its input after being delayed for one clock cycle by the DFF 10. By contrast, the $x(5-0)$ output of the adder logic unit a1 of FIG. 1 is not fed back until after the DFF 50 of FIG. 1. That represents five more clock cycles. Similar to the interleaving approach above, the one delay feedback approach reduces the number of DFFs. In this case, the DFFs are in the input stream. For example, the $x(35-30)$ delayed input of logic unit a6 of FIG. 1 includes five DFFs 0, 11, 21, 30, 38 which are not correspondingly present in FIG. 2. Although each bit segment of the adder section 210 differs, all benefit from the one delay feedback approach in one way or another.

[0001] Also shown in FIG. 2, the averaging filter 200 employs a delay enable signal output approach. This approach would also work with either a subtractor module or an adder module in isolation. According to the delay enable signal output approach, a number of sample-and-hold subsystems En27-En32 are each placed in electrical communication with a corresponding pipelined adder logic unit a1-a5. Each sample-and-hold subsystem samples an output of the corresponding adder logic unit at predetermined time intervals set by a delay enable signal and outputs the sample in the form of a corresponding output after the predetermined delay time. For example, sample-and-hold subsystem En32 samples the output of adder logic unit a1 at predetermined time intervals set by a delay enable signal *Samp1* on input 204a and outputs the sample signal on output 204. By comparison to the adder module 200 of FIG. 1, the delay enable signal approach reduces the number of DFFs needed for processing the output of the adder section 210. For example, the adder

logic unit a1 of FIG. 2 has effectively two DFFs 10, En32. By contrast, the adder logic unit a1 of FIG. 1 has six DFFs 10, 20, 29, 37, 44, 50. Although each bit segment of the adder section 210 differs, most benefit from the delay enable signal output approach.

[0001] Turning now to FIG. 3, a timing diagram illustrating six delay enable signals in accordance with the present invention is shown. Also shown is the clock (*Clk*) signal for reference purposes. The six delay enable signals *Samp1-Samp6* are input to the six sample-and-hold subsystems En27-En32 of FIG. 2 on their corresponding inputs 204a-209a of FIG. 2. As shown, this sampling is performed by the sequentially delayed sampling signals at predetermined time intervals. The predetermined time intervals and the number of sampling signals varies according to the total number of adder logic units used. In the six adder logic units of the exemplary embodiment shown in FIG. 2, the predetermined time intervals in each of the sampling signals *Samp2* to *Samp6* are cued from the rising edge of *Samp1*, which is also used in the subsequent processing stages to enable a read of the final filter output values. Signals having inverse logic could also be used. Accordingly, by producing a series of sampling pulses for the sample-and-hold subsystems, an equivalent functionality of the many DFFs used in the prior art is achieved.

[0001] Applied together, the three approaches presented with respect to FIG. 2 achieve a substantial savings. Recall that the conventional first order averaging filter uses 387 DFFs. By contrast, the first order averaging filter in FIG. 2 only uses 123 DFFs. This is less than one third of the DFF count used in the conventional approach. The foregoing approaches therefore advantageously reduce delay-gate count, thereby reducing the associated processing time delay and manufacturing complication and cost for the implementing additional delay flip flops.

[0001] It should be noted that the three approaches presented with respect to FIG. 2 were discussed separately for clarity of description and that they can be incorporated in whole or in part into a single embodiment of the present invention utilizing all or some of these approaches. It should further be noted that the present invention is not limited to averaging filters, either of first order or in general, but can readily be used in conjunction with virtually any devices that utilizes pipelining.

[0001] Other embodiments, features, and advantages of the present invention will be apparent to those skilled in the art from a consideration of the foregoing specification as well as through practice of the invention and alternative embodiments and methods disclosed herein. Therefore, it should be emphasized that the specification and examples are exemplary only, and that the true scope and spirit of the invention is limited only by the following claims.